**IN THE CLAIMS:**

Please amend claims 1, 3, 5, 8, 11, 31, 35, 40, 42, 44, 45, 47-56, 58, 60, 63, 66, 86, 90, 95, 97, 100-102, 104-113, 115, 147, 152, 166, 168, 170, 202, 207, 209, 225, 227, 259, 264, 280, 282, 314, 319, 321, 449, and 450 as indicated below.

A listing of the status of all claims 1-452 in the present patent application is provided below.

1 (Currently Amended). A fast encoder for compressing input data into output compressed data, comprising:

~~at least one~~ a plurality of single-level direct subband transformers, the plurality of single-level direct subband transformers for receiving and transforming input data to produce transformation coefficients;

~~at least one~~ a plurality of processing means, each of the plurality of processing means directly coupled to at least a respective one of the plurality of single-level direct subband transformers, each of the plurality of processing means selected from a group consisting of: pass-through means for lossless processing and quantizer means for lossy processing~~, coupled to at least one of said at least one single-level direct subband transformer~~, for directly receiving and processing ~~the~~ respective transformation coefficients to produce processed transformation coefficients;

~~at least one~~ a plurality of encoding probability estimators, each of the plurality of encoding probability estimators coupled to at least a respective one of ~~said at least one~~ the plurality of processing means, for receiving ~~the~~ respective processed transformation coefficients and estimating probabilities of symbols within contexts of transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;

~~at least one~~ a plurality of entropy encoders, each of the plurality of entropy encoders coupled to at least a respective one of ~~said at least one~~ the plurality of encoding probability estimators, for receiving and entropy encoding ~~the~~ respective processed transformation coefficients using ~~the~~ respective probabilities of symbols within ~~the~~ respective contexts of transformation coefficients to produce encoded data; and

an output compressed buffer coupled to ~~at least one of said at least one~~ the plurality of entropy encoders, for receiving and substantially synchronizing the encoded data ~~with said fast encoder~~ from each of the plurality of entropy encoders to produce output compressed data.

2 (Cancelled).

3 (Currently Amended). The fast encoder of claim 1, further

comprising:

~~at least one single-level direct subband transformer for receiving and transforming input data to produce transformation coefficients;~~

~~at least one processing means selected from a group consisting of: pass through means for lossless processing and quantizer means for lossy processing, coupled to at least one of said at least one single-level direct subband transformer, for receiving and processing the transformation coefficients to produce processed transformation coefficients,~~

~~at least one encoding probability estimator coupled to at least one of said at least one processing means, for receiving the processed transformation coefficients and estimating probabilities of symbols within contexts of transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;~~

~~at least one entropy encoder coupled to at least one of said at least one encoding probability estimator, for receiving and entropy encoding the processed transformation coefficients using the probabilities of symbols within the contexts of transformation coefficients to produce encoded data;~~

at least one synchronization memory coupled to at least one of ~~said at least one~~ <u>the plurality of</u> entropy encoder<u>s</u>, for receiving and substantially synchronizing the encoded data ~~with~~

~~said fast encoder~~ from the at least one of the plurality of entropy encoders to produce synchronized compressed data; ~~and~~

wherein the [~~an~~] output compressed buffer is further coupled to at least one of said at least one synchronization memory, for receiving and buffering synchronized compressed data to produce the output compressed data.

4 (Previously Presented). The fast encoder of claim 1, further comprising:

at least one color space converter for converting an original input image to produce the input data.

5 (Currently Amended). The fast encoder of claim 1, wherein:

a first of said ~~at least one~~ plurality of single-level direct subband transformers is coupled to receive and transform the input data to produce transformation coefficients; and

each other of said ~~at least one~~ plurality of single-level direct subband transformers is coupled to receive and transform selected transformation coefficients to produce transformed transformation coefficients.

6 (Original). The fast encoder of claim 5, wherein selected transformation coefficients are low-pass transformed for one-dimensional input data.

7 (Original). The fast encoder of claim 5, wherein selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

8 (Currently Amended). The fast encoder of claim 1, wherein said ~~at least one~~ plurality of single-level direct subband transformers comprises:

at least one direct filter for horizontal filtering; and

at least one direct filter for vertical filtering.

9 (Previously Presented). The fast encoder of claim 8, wherein said at least one direct filter for horizontal filtering is different from said at least one direct filter for vertical filtering.

10 (Previously Presented). The fast encoder of claim 8, wherein at least one of said at least one direct filter for horizontal filtering and said at least one direct filter for vertical filtering comprises at least one direct non-stationary filter.

11 (Currently Amended). The fast encoder of claim 1, wherein said ~~at least one~~ plurality of single-level direct subband

transformers comprises at least one direct filter for filtering.

12 (Previously Presented). The fast encoder of claim 11, wherein said at least one direct filter comprises at least one direct non-stationary filter.

13 (Previously Presented). The fast encoder of claim 12, wherein said at least one direct non-stationary filter comprises a plurality of serially coupled direct non-stationary filter cells.

14 (Previously Presented). The fast encoder of claim 1, wherein each said direct non-stationary filter cell comprises:

a filter device;

a filter cell input *x* coupled to said filter device;

a filter cell output *y* coupled to said filter device;

a first switch and a second switch coupled to said filter device, having a plurality of controllable positions; and

a clock input *c* coupled to control said first switch and said second switch, for providing a non-stationarity of said direct non-stationary filter cell.

15 (Previously Presented). The fast encoder of claim 14, wherein:

said first switch is in a first position for horizontal filtering of each second pixel and in a second position for horizontal filtering of other pixels; and

said second switch is in a second position for horizontal filtering of each second pixel and in a first position for horizontal filtering of other pixels.

16 (Previously Presented). The fast encoder of claim 14, wherein:

said first switch is in a first position for vertical filtering of each second line and in a second position for vertical filtering of other lines; and

said second switch is in a second position for vertical filtering of each second line and in a first position for vertical filtering of other lines.

17 (Previously Presented). The fast encoder of claim 14, wherein each said direct non-stationary filter further comprises:

a first gain multiplier;

a second gain multiplier; and

a selection switch, having a plurality of positions controlled by said clock input $c$, wherein

an output of said plurality of serially coupled direct non-

stationary filter cells is coupled to an input of said first gain multiplier, for multiplying said output with a first gain number to produce a first result;

an output of said plurality of serially coupled direct non-stationary filter cells is coupled to an input of said second gain multiplier, for multiplying said output with a second gain number to produce a second result;

an output of said direct non-stationary filter is coupled to an output of said first gain multiplier for said selection switch in the first position; and

an output of said direct non-stationary filter is coupled to an output of said second gain multiplier for said selection switch in the second position.

18 (Previously Presented). The fast encoder of claim 14, wherein said filter device comprises:

at least one delay element $z^{-w}$;

a plurality of multipliers $K_1[0]$, $K_1[1]$, ..., $K_1[k-1]$, $K_2[k-1]$, $K_2[k-2]$, ..., $K_2[0]$; and

a plurality of adders,

wherein:

an output of each even indexed said delay element $z^{-w}$ is coupled to an input of subsequent odd indexed said delay element

$z^{-w}$;

an output of each odd indexed said delay element $z^{-w}$ is coupled to an input of subsequent even indexed said delay element $z^{-w}$;

the output of each even indexed said delay element $z^{-w}$ is coupled to an input of at least one of said plurality of multipliers $K_1[0]$, $K_1[1]$, ..., $K_1[k-1]$;

outputs of all said multipliers $K_1[0]$, $K_1[1]$, ..., $K_1[k-1]$ are coupled to inputs of said adders, for adding together all outputs of all said multipliers $K_1[0]$, $K_1[1]$, ..., $K_1[k-1]$ to produce a first result;

inputs of first said adder are coupled to receive and add the first result with said filter cell input x;

an input of first said delay element $z^{-w}$ is coupled to said filter cell input x for said first switch in the first position;

an input of first said delay element $z^{-w}$ is coupled to the output of first said adder for said first switch in the second position;

said filter cell input x and the output of each odd indexed said delay element $z^{-w}$ is coupled to an input of at least one of said plurality of multipliers $K_2[k-1]$, $K_2[k-2]$, ..., $K_2[0]$;

outputs of all said multipliers $K_2[k-1]$, $K_2[k-2]$, ..., $K_2[0]$

are coupled to inputs of said adders, for adding together all outputs of all said multipliers $K_2[k-1]$, $K_2[k-2]$,..., $K_2[0]$ to produce a second result;

inputs of last said adder are coupled to receive and add the second result with the output of last said delay element $z^{-w}$;

said filter cell output y is coupled to the output of last said delay element $z^{-w}$ for said second switch in the first position; and

said filter cell output y is coupled to the output of last said adder for said second switch in the second position.

19 (Previously Presented). The fast encoder of claim 18, wherein at least one of said multipliers $K_1[0]$, $K_1[1]$,..., $K_1[k-1]$, $K_2[k-1]$, $K_2[k-2]$,..., $K_2[0]$ comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

20 (Previously Presented). The fast encoder of claim 18, wherein said filter device further comprises:

a first function $N_1$ means coupled to receive and transform the first result to produce a third result; and

a second function $N_2$ means coupled to receive and transform the second result to produce a fourth result, wherein:

inputs of first said adder are coupled to receive and add the third result with said filter cell input x; and

inputs of last said adder are coupled to receive and add the fourth result with the output of last said delay element $z^{-w}$.

21 (Previously Presented). The fast encoder of claim 20, wherein at least one of said first function $N_1$ means and said second function $N_2$ means comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

22 (Previously Presented). The fast encoder of claim 14, wherein said filter device comprises:

a delay element $z^{-w}$;

a first multiplier and a second multiplier; and

a first adder and a second adder,

wherein:

an input of said first multiplier is coupled to said filter cell input x;

an input of said second multiplier is coupled to an output of said delay element $z^{-w}$;

inputs of said first adder are coupled to receive and add the output of said second multiplier with said filter cell input

x;

an input of said delay element $z^{-w}$ is coupled to said filter cell input x for said first switch in the first position;

an input of said delay element $z^{-w}$ is coupled to the output of said first adder for said first switch in the second position;

inputs of said second adder are coupled to receive and add the output of said first multiplier with the output of said delay element $z^{-w}$;

said filter cell output y is coupled to the output of said delay element $z^{-w}$ for said second switch in the first position; and

said filter cell output y is coupled to the output of said second adder for said second switch in the second position.

23 (Previously Presented). The fast encoder of claim 22, wherein at least one of said first multiplier and said second multiplier comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

24 (Original). The fast encoder of claim 22, wherein said shifting means shifts right data from its input for two bit

positions to produce data at its output.

25 (Original). The fast encoder of claim 23, wherein said shifting means shifts right data from its input for one bit position to produce data at its output.

26 (Previously Presented). The fast encoder of claim 25, wherein:

inputs of said first adder are coupled to receive and subtract the output of said second multiplier from said filter cell input x; and

inputs of said second adder are coupled to receive and subtract the output of said first multiplier from the output of said delay element $z^{-w}$.

27 (Previously Presented). The fast encoder of claim 14, wherein said filter device comprises:

a first delay element $z^{-w}$, a second delay element $z^{-w}$ and a third delay element $z^{-w}$;

a first multiplier, a second multiplier, a third multiplier and a fourth multiplier; and

a first adder, a second adder, a third adder and a fourth adder,

wherein:

an output of said first delay element $z^{-w}$ is coupled to an input of said second delay element $z^{-w}$ and an input of said second multiplier;

an output of said second delay element $z^{-w}$ is coupled to an input of said third delay element $z^{-w}$ and an input of said third multiplier;

an input of said fourth multiplier is coupled to the output of said third delay element $z^{-w}$;

inputs of said second adder are coupled to receive and add the output of said second multiplier with the output of said fourth multiplier;

inputs of said first adder are coupled to receive and add the output of said second adder with said filter cell input x;

an input of said first delay element $z^{-w}$ is coupled to said filter cell input x for said first switch in the first position;

an input of said first delay element $z^{-w}$ is coupled to the output of said first adder for said first switch in the second position;

inputs of said third adder are coupled to receive and add the output of said third multiplier with the output of said first multiplier;

inputs of said fourth adder are coupled to receive and add

the output of said third adder with the output of said third delay element $z^{-w}$;

said filter cell output y is coupled to the output of said third delay element $z^{-w}$ for said second switch in the first position; and

said filter cell output y is coupled to the output of said fourth adder for said second switch in the second position.

28 (Previously Presented). The fast encoder of claim 27, wherein at least one of said first multiplier, said second multiplier, said third multiplier and said fourth multiplier comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

29 (Original). The fast encoder of claim 28, wherein said shifting means shifts right data from its input for four bit positions to produce data at its output.

30 (Previously Presented). The fast encoder of claim 29, wherein:

inputs of said second adder are coupled to receive and subtract the output of said second multiplier from the output of said fourth multiplier; and

inputs of said third adder are coupled to receive and subtract the output of said third multiplier from the output of said first multiplier.

31 (Currently Amended). The fast encoder of claim 1, wherein said ~~at least one~~ plurality of encoding probability estimators comprises at least one adaptive histogram updating means, for updating an adaptive histogram.

32 (Original). The fast encoder of claim 31, wherein said adaptive histogram updating means comprises a low-pass filter for filtering probabilities selected from a group consisting of:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

33 (Original). The fast encoder of claim 32, wherein said adaptive histogram updating means further comprises a dominant pole adapter for adapting a dominant pole of said low-pass filter.

34 (Original). The fast encoder of claim 33, wherein said dominant pole adapter comprises a dominant pole divider for halving a value of the dominant pole in each adaptation cycle.

35 (Currently Amended). The fast encoder of claim 1, wherein said ~~at least one~~ plurality of entropy encoders ~~is a~~ comprises at least one range encoder, comprising a first multiplier for multiplying a prescaled range r with a number $Q(x)$ selected from a group consisting of:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

36 (Original). The fast encoder of claim 35, wherein said first multiplier comprises:

a first simplified multiplier for multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$; and

a first left shifter coupled to said first simplified multiplier, for shifting left the output of said first simplified multiplier for l bit positions.

37 (Original). The fast encoder of claim 35, wherein said first multiplier comprises a first left shifter for shifting left said number $Q(x)$ for l bit positions.

38 (Original). The fast encoder of claim 35, wherein said first multiplier comprises:

a third left shifter comprising:

means for zeroing its output, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1, and

means for shifting left said number $Q(x)$ for one bit position, when said small number V is any odd number higher or equal 3;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for l bit positions.

39 (Original). The fast encoder of claim 35, wherein said first multiplier comprises:

a third left shifter comprising:

means for zeroing its output, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1,

means for shifting left said number $Q(x)$ for one bit position, when said small number V is equal 3, and

means for shifting left said number $Q(x)$ for two bit

positions, when said small number V is any odd number higher or equal 5;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for 1 bit positions.

40 (Currently Amended). The fast encoder of claim 1, wherein said ~~at least one~~ plurality of entropy encoders ~~is a~~ comprises at least one range encoder, comprising a first divider for dividing a range R with a number Total of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

41 (Original). The fast encoder of claim 40, wherein said first divider comprises a first right shifter for shifting right said range R for $w_3 = \log_2(Total)$ bit positions.

42 (Currently Amended). The fast encoder of claim 1, wherein said ~~at least one~~ plurality of encoding probability estimators comprises:

at least one transformation coefficient C splitter, for

splitting a transformation coefficient *C* into a sign S and a magnitude M;

at least one magnitude-set index MS determinator coupled to said transformation coefficient C splitter, for determining ~~the~~ a magnitude-set index MS using said magnitude M and a magnitude-set table;

at least one residual R determinator, coupled to said transformation coefficient C splitter, for determining a residual R using said magnitude M and said magnitude-set table.

43 (Original). The fast encoder of claim 42, wherein:

said magnitude-set index MS is determined to be equal to a sum of a doubled position of the first nonzero bit of the highest significance and the value of the first next bit of the lower significance in a binary representation of said magnitude M; and

said residual R is determined as the difference between said magnitude M and the lower coefficient limit, equal to a value of said magnitude M with all bits zeroed except the first nonzero bit of the highest significance and the first next bit of the lower significance in a binary representation of said magnitude M.

44 (Currently Amended). The fast encoder of claim 42, wherein

said plurality of entropy encoders comprises at least one residual R encoder coupled to said residual R determinator, for encoding the residual R using variable length coding (VLC).

45 (Currently Amended). The fast encoder of claim 42, wherein said plurality of encoding probability estimators further comprises at least one context modeler of a transformation coefficient to be encoded, using already encoded transformation coefficients.

46 (Original). The fast encoder of claim 45, wherein said already encoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be encoded.

47 (Currently Amended). The fast encoder of claim 45, wherein said plurality of encoding probability estimators further comprises at least one mean value $\overline{MS}$ determinator coupled to said context modeler, for determining $\overline{MS}$ as the mean value of magnitude-set indexes $MS_i$ of said already encoded transformation coefficients.

48 (Currently Amended). The fast encoder of claim 47, wherein

said plurality of encoding probability estimators further comprises at least one maximum mean value $\overline{MS}$ limiter coupled to said mean value $\overline{MS}$ determinator, for limiting a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

49 (Currently Amended). The fast encoder of claim 48, wherein said plurality of entropy encoders comprises at least one magnitude range encoder coupled to said maximum mean value $\overline{MS}$ limiter, for encoding said magnitude-set index MS as a current symbol x, using an adaptive magnitude histogram $h[MC]$.

50 (Currently Amended). The fast encoder of claim 48, wherein said plurality of encoding probability estimators further comprises at least one adaptive magnitude histogram $h[MC]$ updating means coupled to said maximum mean value $\overline{MS}$ limiter, for an adaptive magnitude histogram $h[MC]$ updating using said magnitude-set index MS as a current symbol x.

51 (Currently Amended). The fast encoder of claim 45, wherein said plurality of encoding probability estimators further comprises at least one ternary context TC determinator coupled to said transformation coefficient C splitter, for determining a

ternary context TC as the ternary code of sign values $S_i$ of already encoded transformation coefficients.

52 (Currently Amended). The fast encoder of claim 51, wherein said plurality of encoding probability estimators further comprises at least one sign inverter coupled to said ternary context TC determinator, for inverting less probable said sign S using NEG table.

53 (Currently Amended). The fast encoder of claim 51, wherein said plurality of encoding probability estimators further comprises at least one ternary context TC translator coupled to said ternary context TC determinator, for translating said ternary context TC into a sign context SC using CTX table.

54 (Currently Amended). The fast encoder of claim 53, wherein said plurality of entropy encoders comprises at least one sign range encoder coupled to said ternary context TC translator, for encoding said sign S as a current symbol x, using an adaptive sign histogram $g[SC]$.

55 (Currently Amended). The fast encoder of claim 53, wherein said plurality of encoding probability estimators further

comprises [[~~an~~]] at least one adaptive sign histogram $g[SC]$ updating means coupled to said ternary context TC translator, for an adaptive sign histogram $g[SC]$ updating using said sign S as a current symbol x.

56 (Currently Amended). A fast decoder for decompressing input compressed data into output data, comprising:

an input compressed buffer, for receiving and substantially synchronizing input compressed data ~~with said fast decoder~~ to produce synchronized compressed data;

~~at least one~~ a plurality of entropy decoders, each of the plurality of entropy decoders coupled to said input compressed buffer, for receiving and decoding ~~the~~ respective synchronized compressed data using probabilities of symbols within contexts of transformation coefficients to produce transformation coefficients;

~~at least one~~ a plurality of decoding probability estimators, each of the plurality of decoding probability estimators coupled to at least a respective one of ~~said at least one~~ the plurality of entropy decoders, for receiving ~~the~~ respective transformation coefficients and estimating the probabilities of symbols within the contexts of transformation coefficients to produce the probabilities of symbols within the

contexts of transformation coefficients;

~~at least one~~ a plurality of processing means, each of the plurality of processing means coupled to at least a respective one of the plurality of entropy decoders, each of the plurality of processing means selected from a group consisting of: pass-through means for lossless processing and dequantizer means for lossy processing, ~~coupled to at least one of said at least one entropy decoder,~~ for receiving and processing ~~the~~ respective transformation coefficients to produce processed transformation coefficients; and

~~at least one~~ a plurality of single-level inverse direct subband transformers, the plurality of single-level inverse direct subband transformers directly coupled to at least a respective one of ~~said at least one~~ the plurality of processing means, for receiving and transforming ~~the~~ processed transformation coefficients to produce output data.

57 (Cancelled).

58 (Currently Amended). The fast decoder of claim 56, further comprising:

at least one synchronization memory coupled to said input compressed buffer, for receiving and substantially synchronizing buffered compressed data ~~with said fast decoder~~ to produce

synchronized compressed data;

wherein the [[~~an~~]] input compressed buffer~~, for receiving and buffering~~ is further configured to receive and buffer input compressed data to produce buffered compressed data~~;~~

~~at least one entropy decoder coupled to at least one of said at least one synchronization memory, for receiving and decoding the synchronized compressed data using probabilities of symbols within contexts of transformation coefficients to produce transformation coefficients;~~

~~at least one decoding probability estimator coupled to at least one of said at least one entropy decoder, for receiving the transformation coefficients and estimating the probabilities of symbols within contexts of transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;~~

~~at least one processing means, selected from a group consisting of: pass-through means for lossless processing and dequantizer means for lossy processing, coupled to at least one of said at least one entropy decoder, for receiving and processing the transformation coefficients to produce processed transformation coefficients; and~~

~~at least one single-level inverse subband transformer coupled to at least one of said at least one processing means, for receiving and transforming the processed transformation~~

~~coefficients to produce output data~~.

59 (Original). The fast decoder of claim 56, further comprising at least one color space converter for converting the output data to produce converted output data.

60 (Currently Amended). The fast decoder of claim 56, wherein:

a last of said plurality of single-level inverse subband transformers is coupled to receive and transform transformation coefficients to produce the output data; and

each other of said plurality of single-level inverse subband transformers is coupled to receive and transform transformation coefficients to produce selected transformation coefficients.

61 (Original). The fast decoder of claim 60, wherein said selected transformation coefficients are low-pass transformed for one-dimensional output data.

62 (Original). The fast decoder of claim 60, wherein said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional output data.

63 (Currently Amended). The fast decoder of claim 56, wherein said plurality of single-level inverse subband transformers comprises:

at least one inverse filter for horizontal filtering; and

at least one inverse filter for vertical filtering.

64 (Original). The fast decoder of claim 63, wherein said inverse filter for horizontal filtering is different from said inverse filter for vertical filtering.

65 (Original). The fast decoder of claim 63, wherein at least one of said inverse filter for horizontal filtering and said inverse filter for vertical filtering comprises at least one inverse non-stationary filter.

66 (Currently Amended). The fast decoder of claim 56, wherein said plurality of single-level inverse subband transformers comprises at least one inverse filter for filtering.

67 (Original). The fast decoder of claim 66, wherein said inverse filter comprises at least one inverse non-stationary filter.

68 (Original). The fast decoder of claim 67, wherein said

inverse non-stationary filter comprises a plurality of serially coupled inverse non-stationary filter cells.

69 (Previously Presented). The fast decoder of claim 56, wherein said inverse non-stationary filter cell comprises:

a filter device;

a filter cell input *x* coupled to said filter device;

a filter cell output *y* coupled to said filter device;

a first switch and a second switch coupled to said filter device, having a plurality of controllable positions; and

a clock input *c* coupled to control said first switch and said second switch, for providing a non-stationarity of said direct non-stationary filter cell.

70 (Currently Amended). The fast decoder of claim 69, wherein:

said first switch is in ~~the~~ a second position for ~~the~~ horizontal filtering of each second pixel and in ~~the~~ a first position for ~~the~~ horizontal filtering of other pixels; and

said second switch is in ~~the~~ a first position for ~~the~~ horizontal filtering of each second pixel and in ~~the~~ a second position for ~~the~~ horizontal filtering of other pixels.

71 (Currently Amended). The fast decoder of claim 69,

wherein:

said first switch is in ~~the~~ a second position for ~~the~~ vertical filtering of each second line and in ~~the~~ a first position for ~~the~~ vertical filtering of other lines; and

said second switch is in ~~the~~ a first position for ~~the~~ vertical filtering of each second line and in ~~the~~ a second position for ~~the~~ vertical filtering of other lines.

72 (Previously Presented). The fast decoder of claim 69, wherein said direct non-stationary filter further comprises:

a first gain multiplier;

a second gain multiplier; and

a selection switch, having a plurality of positions controlled by said clock input $c$, wherein:

an input of said first gain multiplier is coupled to an input of said inverse non-stationary filter, for multiplying an input sample with a reciprocal value of a first gain number to produce a first result;

an input of said second gain multiplier is coupled to an input of said inverse non-stationary filter, for multiplying an input sample with a reciprocal value of a second gain number to produce a second result;

an input of said plurality of serially coupled inverse non-stationary filter cells is coupled to an output of said first

gain multiplier, for said selection switch in the second position; and

an input of said plurality of serially coupled inverse non-stationary filter cells is coupled to an output of said second gain multiplier, for said selection switch in the first position.

73 (Previously Presented). The fast decoder of claim 69, wherein said filter device comprises:

at least one delay element $z^{-w}$;

a plurality of multipliers $K_1[0]$, $K_1[1]$, ..., $K_1[k-1]$, $K_2[k-1]$, $K_2[k-2]$, ..., $K_2[0]$; and

a plurality of adders,

wherein:

an output of each even indexed said delay element $z^{-w}$ is coupled to an input of subsequent odd indexed said delay element $z^{-w}$;

an output of each odd indexed said delay element $z^{-w}$ is coupled to an input of subsequent even indexed said delay element $z^{-w}$;

the output of each even indexed said delay element $z^{-w}$ is coupled to an input of at least one of said plurality of

multipliers $K_1[0]$, $K_1[1]$, . . ., $K_1[k-1]$;

outputs of all said multipliers $K_1[0]$, $K_1[1]$, . . ., $K_1[k-1]$ are coupled to inputs of said adders, for adding together all outputs of all said multipliers $K_1[0]$, $K_1[1]$, . . ., $K_1[k-1]$ to produce a first result;

inputs of first said adder are coupled to receive and add the first result with said filter cell input x;

an input of first said delay element $z^{-w}$ is coupled to said filter cell input x for said first switch in the first position;

an input of first said delay element $z^{-w}$ is coupled to the output of first said adder for said first switch in the second position;

said filter cell input x and the output of each odd indexed said delay element $z^{-w}$ is coupled to an input of at least one of said plurality of multipliers $K_2[k-1]$, $K_2[k-2]$, . . ., $K_2[0]$;

outputs of all said multipliers $K_2[k-1]$, $K_2[k-2]$, . . ., $K_2[0]$ are coupled to inputs of said adders, for adding together all outputs of all said multipliers $K_2[k-1]$, $K_2[k-2]$, . . ., $K_2[0]$ to produce a second result;

inputs of last said adder are coupled to receive and add the second result with the output of last said delay element $z^{-w}$;

said filter cell output y is coupled to the output of last

said delay element $z^{-w}$ for said second switch in the first position; and

said filter cell output y is coupled to the output of last said adder for said second switch in the second position.

74 (Previously Presented). The fast decoder of claim 73, wherein at least one of said multipliers $K_1[0]$, $K_1[1]$, ..., $K_1[k-1]$, $K_2[k-1]$, $K_2[k-2]$, ..., $K_2[0]$ comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

75 (Previously Presented). The fast decoder of claim 73, wherein said inverse non-stationary filter cell further comprises:

a first function $N_1$ means coupled to receive and transform the first result to produce a third result; and

a second function $N_2$ means coupled to receive and transform the second result to produce a fourth result,

wherein:

inputs of first said adder are coupled to receive and add the third result with said filter cell input x; and

inputs of last said adder are coupled to receive and add the fourth result with the output of last said delay element $z^{-w}$.

76 (Previously Presented). The fast decoder of claim 75, wherein at least one of said first function $N_1$ means and said second function $N_2$ means comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

77 (Previously Presented). The fast decoder of claim 69, wherein said filter device comprises:

a delay element $z^{-w}$;

a first multiplier and a second multiplier; and

a first adder and a second adder,

wherein:

an input of said first multiplier is coupled to said filter cell input x;

an input of said second multiplier is coupled to an output of said delay element $z^{-w}$;

inputs of said first adder are coupled to receive and add the output of said second multiplier with said filter cell input x;

an input of said delay element $z^{-w}$ is coupled to said filter cell input x, for said first switch in the first position;

an input of said delay element $z^{-w}$ is coupled to the output

of said first adder, for said first switch in the second position;

inputs of said second adder are coupled to receive and add the output of said first multiplier with the output of said delay element $z^{-w}$;

said filter cell output y is coupled to the output of said delay element $z^{-w}$, for said second switch in the first position; and

said filter cell output y is coupled to the output of said second adder, for said second switch in the second position.

78 (Previously Presented). The fast decoder of claim 77, wherein at least one of said first multiplier and said second multiplier comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

79 (Original). The fast decoder of claim 78, wherein said shifting means shifts right data from its input for one bit position to produce data at its output.

80 (Original). The fast decoder of claim 78, wherein said shifting means shifts right data from its input for two bit

positions to produce data at its output.

81 (Previously Presented). The fast decoder of claim 80, wherein:

inputs of said first adder are coupled to receive and subtract the output of said second multiplier from said filter cell input x; and

inputs of said second adder are coupled to receive and subtract the output of said first multiplier from the output of said delay element $z^{-w}$.

82 (Previously Presented). The fast decoder of claim 69, wherein said filter device comprises:

a first delay element $z^{-w}$, a second delay element $z^{-w}$ and a third delay element $z^{-w}$;

a first multiplier, a second multiplier, a third multiplier and a fourth multiplier; and

a first adder, a second adder, a third adder and a fourth adder,

wherein:

an output of said first delay element $z^{-w}$ is coupled to an input of said second delay element $z^{-w}$ and an input of said second multiplier;

an output of said second delay element $z^{-w}$ is coupled to an input of said third delay element $z^{-w}$ and an input of said third multiplier;

an input of said fourth multiplier is coupled to the output of said third delay element $z^{-w}$;

inputs of said second adder are coupled to receive and add the output of said fourth multiplier with the output of said second multiplier;

inputs of said first adder are coupled to receive and add the output of said second adder with said filter cell input x;

an input of said first delay element $z^{-w}$ is coupled to said filter cell input x, for said first switch in the first position;

an input of said first delay element $z^{-w}$ is coupled to the output of said first adder, for said first switch in the second position;

inputs of said third adder are coupled to receive and add the output of said first multiplier with the output of said third multiplier;

inputs of said fourth adder are coupled to receive and add the output of said third adder with the output of said third delay element $z^{-w}$;

said filter cell output y is coupled to the output of said

third delay element $z^{-w}$, for said second switch in the first position; and

said filter cell output y is coupled to the output of said fourth adder, for said second switch in the second position.

83 (Previously Presented). The fast decoder of claim 82, wherein at least one of said first multiplier, said second multiplier, said third multiplier and said fourth multiplier comprises a shifting means selected from a group consisting of: shifters and shifted hardwired bit line connections.

84 (Original). The fast decoder of claim 83, wherein said shifting means shifts right data from its input for four bit positions to produce data at its output.

85 (Previously Presented). The fast decoder of claim 84, wherein:

inputs of said second adder are coupled to receive and subtract the output of said fourth multiplier from the output of said second multiplier; and

inputs of said third adder are coupled to receive and subtract the output of said first multiplier from the output of said third multiplier.

86 (Currently Amended). The fast decoder of claim 56, wherein said plurality of decoding probability estimators comprises at least one adaptive histogram updating means, for updating an adaptive histogram.

87 (Original). The fast decoder of claim 86, wherein said adaptive histogram updating means comprises a low-pass filter for filtering probabilities selected from a group consisting of:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

88 (Original). The fast decoder of claim 87, wherein said adaptive histogram updating means further comprises a dominant pole adapter for adapting a dominant pole of said low-pass filter.

89 (Original). The fast decoder of claim 88, wherein said dominant pole adapter comprises a dominant pole divider for halving a value of the dominant pole in each adaptation cycle.

90 (Currently Amended). The fast decoder of claim 56, wherein said plurality of entropy decoders ~~is a~~ comprises at least one range decoder, comprising a first multiplier for multiplying a

prescaled range r with a number $Q(x)$ selected from a group consisting of:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

91 (Original). The fast decoder of claim 90, wherein said first multiplier comprises:

a first simplified multiplier for multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$; and

a first left shifter coupled to said first simplified multiplier, for shifting left the output of said first simplified multiplier for l bit positions.

92 (Original). The fast decoder of claim 90, wherein said first multiplier comprises a first left shifter for shifting left said number $Q(x)$ for l bit positions.

93 (Original). The fast decoder of claim 90, wherein said first multiplier comprises:

a third left shifter comprising:

means for zeroing its output, when a small number

$V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1, and

means for shifting left said number $Q(x)$ for one bit position, when said small number V is any odd number higher or equal 3;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for l bit positions.

94 (Original). The fast decoder of claim 90, wherein said first multiplier comprises:

a third left shifter comprising:

means for zeroing its output, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1,

means for shifting left said number $Q(x)$ for one bit position, when said small number V is equal 3, and

means for shifting left said number $Q(x)$ for two bit positions, when said small number V is any odd number higher or equal 5;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left

shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for l bit positions.

95 (Currently Amended). The fast decoder of claim 56, wherein said plurality of entropy decoders ~~is a~~ comprises at least one range decoder, comprising a first divider for dividing a range R with a number Total of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

96 (Original). The fast decoder of claim 95, wherein said first divider comprises a first right shifter for shifting right said range R for $w_3 = \log_2(Total)$ bit positions.

97 (Currently Amended). The fast decoder of claim 56, wherein said plurality of entropy decoders ~~is a~~ comprises at least one range decoder, comprising a second divider for dividing a bottom range limit B with a prescaled range r, to produce a range correction $t = \lfloor B/r \rfloor$.

98 (Original). The fast decoder of claim 97, wherein said second divider comprises:

a second simplified divider for dividing said bottom range limit B with a small number $V = \lfloor r \cdot 2^{-l} \rfloor$; and

a second right shifter coupled to said second simplified divider, for shifting right the output of said second simplified divider for l bit positions.

99 (Original). The fast decoder of claim 97, wherein said second divider comprises:

a third multiplier for multiplying said bottom range limit B with a first predefined number, dependent on a small number $V = \lfloor r \cdot 2^{-l} \rfloor$; and

a second right shifter coupled to said third multiplier, for shifting right the output of said third multiplier for a sum of l and a second predefined number of bit positions, dependent on said small number V.

100 (Currently Amended). The fast decoder of claim 56, wherein said <u>plurality of</u> decoding probability estimator<u>s</u> comprises a<u>t</u> <u>least one</u> transformation coefficient C builder for reconstructing transformation coefficient C using a magnitude-set index MS, a sign S and a residual R.

101 (Currently Amended). The fast decoder of claim 100,

wherein said plurality of entropy decoders comprises at least one residual R decoder, for decoding said residual R using inverse variable length coding (INVVLC).

102 (Currently Amended). The fast decoder of claim 100, wherein said plurality of decoding probability estimators further comprises at least one context modeler of a transformation coefficient to be decoded, using already decoded transformation coefficients.

103 (Original). The fast decoder of claim 102, wherein said already decoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be decoded.

104 (Currently Amended). The fast decoder of claim 102, wherein said plurality of decoding probability estimators further comprises at least one mean value $\overline{MS}$ determinator coupled to said context modeler, for determining $\overline{MS}$ as the mean value of magnitude-set indexes $MS_i$ of said already decoded transformation coefficients.

105 (Currently Amended). The fast decoder of claim 104,

wherein said plurality of decoding probability estimators further comprises at least one maximum mean value $\overline{MS}$ limiter coupled to said mean value $\overline{MS}$ determinator, for limiting a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

106 (Currently Amended). The fast decoder of claim 105, wherein said plurality of entropy decoders comprises at least one magnitude range decoder coupled to said maximum mean value $\overline{MS}$ limiter, for decoding a magnitude-set index MS as a current symbol x, using an adaptive magnitude histogram $h[MC]$.

107 (Currently Amended). The fast decoder of claim 106, wherein said plurality of decoding probability estimators further comprises at least one adaptive magnitude histogram $h[MC]$ updating means coupled to said magnitude range decoder, for an adaptive magnitude histogram $h[MC]$ updating using decoded said magnitude-set index MS as a current symbol x.

108 (Currently Amended). The fast decoder of claim 102, wherein said plurality of decoding probability estimators further comprises at least one ternary context TC determinator coupled to said context modeler, for determining a ternary

context TC as the ternary code of sign values $S_i$ of already decoded transformation coefficients.

109 (Currently Amended). The fast decoder of claim 108, wherein said plurality of decoding probability estimators further comprises at least one ternary context TC translator coupled to said ternary context TC determinator, for translating ternary context TC into a sign context SC using CTX table.

110 (Currently Amended). The fast decoder of claim 109, wherein said plurality of entropy decoders comprises at least one sign range decoder coupled to said ternary context TC translator, for decoding a sign S as a current symbol x, using an adaptive sign histogram $g[SC]$.

111 (Currently Amended). The fast decoder of claim 110, wherein said plurality of decoding probability estimators further comprises at least one adaptive sign histogram $g[SC]$ updating means coupled to said sign range decoder, for an adaptive sign histogram $g[SC]$ updating using decoded said sign S as a current symbol x.

112 (Currently Amended). The fast decoder of claim 110,

wherein said plurality of decoding probability estimators further comprises at least one sign inverter coupled to said sign range decoder, for inverting less probable said sign S using NEG table.

113 (Currently Amended). A method for fast encoding input data into output compressed data, comprising:

receiving input data at a plurality of single-level direct subband transformers;

~~direct subband~~ transforming the input data received at the plurality of single-level direct subband transformers to produce transformation coefficients;

receiving respective transformation coefficients directly from the plurality of single-level direct subband transformers at a plurality of processing means;

processing the respective transformation coefficients received at each of the plurality of processing means ~~selected from a group consisting of: passing-through transformation coefficients for lossless processing and quantizing transformation coefficients for lossy processing~~ to produce processed transformation coefficients, each of the plurality of processing means selected from a group consisting of: pass-through means for lossless processing and quantizer means for lossy processing;

estimating probabilities of symbols within contexts of transformation coefficients using the processed transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;

entropy encoding the processed transformation coefficients using the probabilities of symbols within the contexts of transformation coefficients to produce encoded data; and

substantially synchronizing the encoded data to produce output compressed data.

114 (Cancelled).

115 (Currently Amended). The method for fast encoding of claim 113, further comprising:

~~direct subband transforming input data to produce transformation coefficients;~~

~~processing transformation coefficients selected from a group consisting of: passing-through transformation coefficients for lossless processing and quantizing transformation coefficients for lossy processing to produce processed transformation coefficients;~~

~~estimating probabilities of symbols within contexts of transformation coefficients using the processed transformation coefficients to produce the probabilities of symbols within~~

~~contexts of transformation coefficients;~~

~~entropy encoding processed transformation coefficients using the probabilities of symbols within contexts of transformation coefficients to produce encoded data;~~

substantially synchronizing the encoded data to produce synchronized compressed data; and

buffering the synchronized compressed data to produce the output compressed data.

116 (Original). The method for fast encoding of claim 113, further comprising color space converting original input data to produce input data.

117 (Original). The method for fast encoding of claim 113, wherein said direct subband transforming comprises:

(a) direct subband transforming the input data to produce transformation coefficients;

(b) direct subband transforming selected transformation coefficients to produce transformed transformation coefficients; and

(c) repeating step (b) finite number of times.

118 (Original). The method for fast encoding of claim 117, wherein said selected transformation coefficients are low-pass

transformed for one-dimensional input data.

119 (Original). The method for fast encoding of claim 117, wherein said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

120 (Original). The method for fast encoding of claim 113, wherein said direct subband transforming comprises:

at least one horizontal direct filtering; and

at least one vertical direct filtering.

121 (Original). The method for fast encoding of claim 120, wherein said horizontal direct filtering is different from said vertical direct filtering.

122 (Original). The method for fast encoding of claim 120, wherein at least one of said horizontal direct filtering and said vertical direct filtering comprises at least one direct non-stationary filtering.

123 (Original). The method for fast encoding of claim 113, wherein said direct subband transforming comprises at least one direct filtering.

124 (Original). The method for fast encoding of claim 123, wherein said direct filtering comprises at least one direct non-stationary filtering.

125 (Original). The method for fast encoding of claim 124, wherein said direct non-stationary filtering comprises a plurality of successive direct non-stationary cell filtering steps.

126 (Currently Amended). The method for fast encoding of claim 125, wherein said direct non-stationary cell filtering comprises:

filtering using a first direct transfer function in ~~the~~ a first cycle; and

filtering using a second direct transfer function in ~~the~~ a second cycle.

127 (Previously Presented). The method for fast encoding of claim 126, wherein:

said first cycle is active during horizontal filtering of each second pixel; and

said second cycle is active during horizontal filtering of other pixels.

128 (Previously Presented). The method for fast encoding of claim 126, wherein:

said first cycle is active during vertical filtering of each second line; and

said second cycle is active during vertical filtering of other lines.

129 (Original). The method for fast encoding of claim 126, further comprising:

first multiplying a result of said plurality of successive direct non-stationary cell filtering steps with a first gain number to produce a first result;

second multiplying a result of said plurality of successive direct non-stationary cell filtering steps with a second gain number to produce a second result;

selecting the first result in each first cycle to produce an output sample; and

selecting the second result in each second cycle to produce the output sample.

130 (Previously Presented). The method for fast encoding of claim 126, wherein said direct non-stationary cell filtering further comprises:

delaying an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiplying each even indexed delayed result with at least one first filter coefficient selected from a group of first filter coefficients to produce first results;

adding together all first results to produce a third result;

adding the third result with the input sample to produce a fifth result;

delaying the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiplying the input sample and each odd indexed delayed result with at least one second filter coefficient selected from a group of second filter coefficients to produce second results;

adding together all second results to produce a fourth result;

adding the fourth result with last delayed result to produce a sixth result;

outputting the sixth result in each first cycle; and

outputting the last delayed result in each second cycle.

131 (Original). The method for fast encoding of claim 130,

wherein at least one said multiplying comprises an operation selected from a group consisting of: shifting and bit remapping.

132 (Original). The method for fast encoding of claim 130, wherein said direct non-stationary cell filtering further comprises:

transforming the third result by first function $N_1$ to produce a seventh result;

transforming the fourth result by first function $N_2$ to produce an eight result;

adding the seventh result with the input sample to produce a fifth result; and

adding the eight result with last delayed result to produce a sixth result.

133 (Original). The method for fast encoding of claim 132, wherein at least one said transforming comprises an operation selected from a group consisting of: shifting and bit remapping.

134 (Original). The method for fast encoding of claim 126, wherein said direct non-stationary cell filtering further comprises:

delaying an input sample for w samples to produce a delayed

result in each first cycle;

second multiplying the delayed result with a second filter coefficient to produce a second result;

first adding the second result with the input sample to produce a fourth result;

delaying the fourth result for w samples to produce the delayed result in each second cycle;

first multiplying the input sample with a first filter coefficient to produce a first result;

second adding the first result with the delayed result to produce a third result;

outputting the third result in each first cycle; and

outputting the delayed result in each second cycle.

135 (Original). The method for fast encoding of claim 134, wherein at least one of said first multiplying and said second multiplying comprises an operation selected from a group consisting of: shifting and bit remapping.

136 (Original). The method for fast encoding of claim 135, wherein said operation comprises shifting right for two bit positions.

137 (Original). The method for fast encoding of claim 135,

wherein said operation comprises shifting right for one bit position.

138 (Previously Presented). The method for fast encoding of claim 137, wherein:

said first adding comprises subtracting the second result from the input sample to produce a fourth result; and

said second adding comprises subtracting the first result from the delayed result to produce a third result.

139 (Original). The method for fast encoding of claim 126, wherein said direct non-stationary cell filtering further comprises:

delaying the input sample for w samples to produce a first delayed result in each first cycle;

delaying the first delayed result for w samples to produce a second delayed result;

delaying the second delayed result for w samples to produce a third delayed result;

first multiplying the input sample with a first filter coefficient to produce a first result;

second multiplying the first delayed result with a second filter coefficient to produce a second result;

third multiplying the second delayed result with a third

filter coefficient to produce a third result;

fourth multiplying the third delayed result with a fourth filter coefficient to produce a fourth result;

second adding the second result with the fourth result to produce a sixth result;

third adding the third result with the first result to produce a fifth result;

first adding the sixth result with the input sample to produce an eight result;

delaying the eight result for w samples to produce the first delayed result in each second cycle;

fourth adding the fifth result with the third delayed result to produce a seventh result;

outputting the seventh result in each first cycle; and

outputting the third delayed result in each second cycle.

140 (Original). The method for fast encoding of claim 139, wherein at least one of said first multiplying, said second multiplying, said third multiplying and said fourth multiplying comprises an operation selected from a group consisting of: shifting and bit remapping.

141 (Original). The method for fast encoding of claim 140, wherein said operation comprises shifting right for four bit

positions.

142 (Original). The method for fast encoding of claim 141, wherein:

said second adding comprises subtracting the second result from the fourth result to produce a sixth result; and

said third adding comprises subtracting the third result from the first result to produce a fifth result.

143 (Previously Presented). The method for fast encoding of claim 113, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients comprises updating adaptive histograms.

144 (Original). The method for fast encoding of claim 143, wherein said updating adaptive histograms comprises low-pass filtering probabilities selected from a group consisting of:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

145 (Original). The method for fast encoding of claim 144, wherein said updating adaptive histograms further comprises adapting a dominant pole during said low-pass filtering.

146 (Original). The method for fast encoding of claim 145, wherein said adapting a dominant pole comprises halving value of the dominant pole in each adaptation cycle.

147 (Currently Amended). The method for fast encoding of claim 113, wherein said entropy encoding ~~is~~ comprises range encoding, comprising first multiplying a prescaled range r with a number $Q(x)$ selected from a group consisting of:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

148 (Original). The method for fast encoding of claim 147, wherein said first multiplying comprises:

simplified multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shifting left the first result for l bit positions.

149 (Original). The method for fast encoding of claim 147, wherein said first multiplying comprises shifting left said number $Q(x)$ for l bit positions.

150 (Original). The method for fast encoding of claim 147, wherein said first multiplying comprises:

zeroing a first result, when a small number $V=\lfloor r\cdot 2^{-l}\rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for l bit positions.

151 (Original). The method for fast encoding of claim 147, wherein said first multiplying comprises:

zeroing a first result, when a small number $V=\lfloor r\cdot 2^{-l}\rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shifting left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for 1 bit positions.

152 (Currently Amended). The method for fast encoding of claim 113, wherein said entropy encoding ~~is~~ comprises range encoding, comprising first dividing a range R with a number Total of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

153 (Original). The method for fast encoding of claim 152, wherein said first dividing comprises shifting right said range R for $w_3 = \log_2(Total)$ bit positions.

154 (Previously Presented). The method for fast encoding of claim 113, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients comprises:

splitting a transformation coefficient *C* into a sign *S* and a magnitude *M*;

determining a magnitude-set index *MS* using said magnitude *M* and a magnitude-set table; and

determining a residual *R* using said magnitude *M* and said magnitude-set table.

155 (Previously Presented). The method for fast encoding of claim 154, wherein:

said magnitude-set index *MS* is determined to be equal to a sum of a doubled position of the first nonzero bit of the highest significance and the value of the first next bit of the lower significance in a binary representation of said magnitude *M*; and

said residual *R* is determined as the difference between said magnitude *M* and the lower coefficient limit, equal to a value of said magnitude *M* with all bits zeroed except the first nonzero bit of the highest significance and the first next bit of the lower significance in a binary representation of said magnitude *M*.

156 (Original). The method for fast encoding of claim 154, wherein said entropy encoding comprises encoding a residual *R* using variable length coding (VLC).

157 (Previously Presented). The method for fast encoding of claim 154, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises context modeling a transformation coefficient to be encoded, using already encoded transformation coefficients.

158 (Original). The method for fast encoding of claim 157, wherein said already encoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be encoded.

159 (Previously Presented). The method for fast encoding of claim 157, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises determining a mean value $\overline{MS}$ as the mean value of magnitude-set indexes $MS_i$ of said already encoded transformation coefficients.

160 (Previously Presented). The method for fast encoding of claim 159, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises limiting a maximum mean value $\overline{MS}$ by a constant $ML$ to produce a magnitude context $MC$.

161 (Original). The method for fast encoding of claim 160, wherein said entropy encoding comprises range encoding said magnitude-set index $MS$ as a current symbol $x$, using an adaptive magnitude histogram $h[MC]$.

162 (Previously Presented). The method for fast encoding of claim 160, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises updating of an adaptive magnitude histogram $h[MC]$ using said magnitude-set index *MS* as a current symbol $x_i$.

163 (Previously Presented). The method for fast encoding of claim 162, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises determining a ternary context *TC* as the ternary code of sign values $S_i$ of said already encoded transformation coefficients.

164 (Previously Presented). The method for fast encoding of claim 163, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises inverting less probable said sign *S* using NEG table.

165 (Previously Presented). The method for fast encoding of claim 163, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients

further comprises translating said ternary context $TC$ into a sign context $SC$ using CTX table.

166 (Currently Amended). The method for fast encoding of claim 165, wherein said entropy ~~encoder~~ <u>encoding</u> comprises range encoding said sign $S$ as a current symbol $x$, using an adaptive sign histogram $g[SC]$.

167 (Previously Presented). The method for fast encoding of claim 165, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises updating of an adaptive sign histogram $g[SC]$ using said sign $S$ as a current symbol $x$.

168 (Currently Amended). A method for fast decoding of input compressed data into output data, comprising:

substantially synchronizing input compressed data to produce synchronized compressed data;

entropy decoding the synchronized compressed data using probabilities of symbols within contexts of transformation coefficients to produce transformation coefficients;

estimating the probabilities of symbols within the contexts of transformation coefficients using the transformation

coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;

processing the transformation coefficients in a plurality of processing means ~~selected from a group consisting of: passing through transformation coefficients for lossless processing and dequantizing transformation coefficients for lossy processing~~ to produce processed transformation coefficients, each of the plurality of processing means selected from a group consisting of: pass-through means for lossless processing and dequantizer means for lossy processing; ~~and~~

receiving respective processed transformation coefficients directly from the plurality of processing means at a plurality of single-level inverse direct subband transformers; and

~~inverse subband~~ transforming the respective processed transformation coefficients received at the plurality of single-level inverse direct subband transformers to produce output data.

169 (Cancelled).

170 (Currently Amended). The method for fast decoding of claim 168, further comprising:

buffering input compressed data to produce buffered compressed data; and

substantially synchronizing the buffered compressed data to produce synchronized compressed data~~;~~

~~entropy decoding the synchronized compressed data using probabilities of symbols within contexts of transformation coefficients to produce transformation coefficients;~~

~~estimating the probabilities of symbols within the contexts of transformation coefficients using the transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;~~

~~processing transformation coefficients selected from a group consisting of: passing-through transformation coefficients for lossless processing and dequantizing transformation coefficients for lossy processing to produce processed transformation coefficients; and~~

~~inverse subband transforming the processed transformation coefficients to produce output data~~.

171 (Original). The method for fast decoding of claim 168, further comprising color space converting the output data to produce converted output data.

172 (Original). The method for fast decoding of claim 168, wherein said inverse subband transforming comprises:

(a) inverse subband transforming transformation

coefficients to produce selected transformation coefficients;

(b) repeating step (a) finite number of times; and

(c) inverse subband transforming transformation coefficients to produce the output data.

173 (Original). The method for fast decoding of claim 172, wherein said selected transformation coefficients are low-pass transformed for one-dimensional output data.

174 (Original). The method for fast decoding of claim 172, wherein said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional output data.

175 (Original). The method for fast decoding of claim 168, wherein said inverse subband transforming comprises:

at least one horizontal inverse filtering; and

at least one vertical inverse filtering.

176 (Original). The method for fast decoding of claim 175, wherein said horizontal inverse filtering is different from said vertical inverse filtering.

177 (Original). The method for fast decoding of claim 175,

wherein at least one of said horizontal inverse filtering and said vertical inverse filtering comprises at least one inverse non-stationary filtering.

178 (Original). The method for fast decoding of claim 168, wherein said inverse subband transforming comprises at least one inverse filtering.

179 (Original). The method for fast decoding of claim 178, wherein said inverse filtering comprises at least one inverse non-stationary filtering.

180 (Original). The method for fast decoding of claim 179, wherein said inverse non-stationary filtering comprises a plurality of successive inverse non-stationary cell filtering steps.

181 (Currently Amended). The method for fast decoding of claim 180, wherein said inverse non-stationary cell filtering comprises:

filtering using a first inverse transfer function in ~~the~~ a first cycle; and

filtering using a second inverse transfer function in ~~the~~ a second cycle.

182 (Previously Presented). The method for fast decoding of claim 181, wherein:

said second cycle is active during horizontal filtering of each second pixel; and

said first cycle is active during horizontal filtering of other pixels.

183 (Previously Presented). The method for fast decoding of claim 181, wherein:

said second cycle is active during vertical filtering of each second line; and

said first cycle is active during vertical filtering of other lines.

184 (Original). The method for fast decoding of claim 181, wherein said inverse non-stationary cell filtering further comprising:

first multiplying an input with a reciprocal value of a first gain number to produce a first result;

second multiplying an input with a reciprocal value of the second gain number to produce a second result;

selecting first result in each second cycle to produce an input sample for said plurality of successive inverse non-

stationary cell filtering steps; and

selecting second result in each first cycle to produce the input sample for said plurality of successive inverse non-stationary cell filtering steps.

185 (Previously Presented). The method for fast decoding of claim 181, wherein said inverse non-stationary cell filtering further comprises:

delaying an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiplying each even indexed delayed result with at least one first filter coefficient selected from a group of first filter coefficients to produce first results;

adding together all first results to produce a third result;

adding the third result with the input sample to produce a fifth result;

delaying the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiplying the input sample and each odd indexed delayed result with at least one second filter coefficient selected from the group of second filter coefficients to produce second

results;

adding together all second results to produce a fourth result;

adding the fourth result with last delayed result to produce a sixth result;

outputting the sixth result in each first cycle; and

outputting the last delayed result in each second cycle.

186 (Original). The method for fast decoding of claim 185, wherein at least one said multiplying comprises an operation selected from a group consisting of: shifting and bit remapping.

187 (Original). The method for fast decoding of claim 185, wherein said inverse non-stationary cell filtering further comprises:

transforming the third result by first function $N_1$ to produce a seventh result;

transforming the fourth result by first function $N_2$ to produce an eight result;

adding the seventh result with the input sample to produce a fifth result; and

adding the eight result with last delayed result to produce a sixth result.

188 (Original). The method for fast decoding of claim 187, wherein at least one said transforming comprises an operation selected from a group consisting of: shifting and bit remapping.

189 (Original). The method for fast decoding of claim 181, wherein said inverse non-stationary cell filtering further comprises:

delaying an input sample for w samples to produce a delayed result in each first cycle;

second multiplying the delayed result with a second filter coefficient to produce a second result;

first adding the second result with the input sample to produce a fourth result;

delaying the fourth result for w samples to produce the delayed result in each second cycle;

first multiplying the input sample with a first filter coefficient to produce a first result;

second adding the first result with the delayed result to produce a third result;

outputting the third result in each first cycle; and

outputting the delayed result in each second cycle.

190 (Original). The method for fast decoding of claim 189,

wherein at least one of said first multiplying and said second multiplying comprises an operation selected from a group consisting of: shifting and bit remapping.

191 (Original). The method for fast decoding of claim 190, wherein said operation comprises shifting right for one bit position.

192 (Original). The method for fast decoding of claim 190, wherein said operation comprises shifting right for two bit positions.

193 (Original). The method for fast decoding of claim 192, wherein:

said first adding comprises subtracting the second result from the input sample to produce a fourth result; and

said second adding comprises subtracting the first result from the delayed result to produce a third result.

194 (Original). The method for fast decoding of claim 181, wherein said inverse non-stationary cell filtering further comprises:

delaying the input sample for w samples to produce a first delayed result in each first cycle;

delaying the first delayed result for w samples to produce a second delayed result;

delaying the second delayed result for w samples to produce a third delayed result;

first multiplying the input sample with a first filter coefficient to produce a first result;

second multiplying the first delayed result with a second filter coefficient to produce a second result;

third multiplying the second delayed result with a third filter coefficient to produce a third result;

fourth multiplying the third delayed result with a fourth filter coefficient to produce a fourth result;

second adding the fourth result with the second result to produce a sixth result;

third adding the first result with the third result to produce a fifth result;

first adding the sixth result with the input sample to produce an eight result;

delaying the eight result for w samples to produce the first delayed result in each second cycle;

fourth adding the fifth result with the third delayed result to produce a seventh result;

outputting the seventh result in each first cycle; and

outputting the third delayed result in each second cycle.

195 (Original). The method for fast decoding of claim 194, wherein at least one of said first multiplying, said second multiplying, said third multiplying and said fourth multiplying comprises an operation selected from a group consisting of: shifting and bit remapping.

196 (Original). The method for fast decoding of claim 195, wherein said operation comprises shifting right for four bit positions.

197 (Original). The method for fast decoding of claim 196, wherein:

said second adding comprises subtracting the fourth result from the second result to produce a sixth result; and

said third adding comprises subtracting the first result from the third result to produce a fifth result.

198 (Previously Presented). The method for fast decoding of claim 168, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients comprises updating adaptive histograms.

199 (Original). The method for fast decoding of claim 198,

wherein said updating adaptive histograms comprises low-pass filtering probabilities selected from a group consisting of:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding said current symbol x.

200 (Original). The method for fast decoding of claim 199, wherein said updating adaptive histograms further comprises adapting a dominant pole during said low-pass filtering.

201 (Original). The method for fast decoding of claim 200, wherein said adapting a dominant pole comprises halving value of the dominant pole in each adaptation cycle.

202 (Currently Amended). The method for fast decoding of claim 168, wherein said entropy decoding ~~is~~ <u>comprises</u> range decoding, comprising first multiplying a prescaled range r with a number $Q(x)$ selected from a group consisting of:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

203 (Original). The method for fast decoding of claim 202, wherein said first multiplying comprises:

simplified multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shifting left the first result for 1 bit positions.

204 (Original). The method for fast decoding of claim 202, wherein said first multiplying comprises shifting left said number $Q(x)$ for 1 bit positions.

205 (Original). The method for fast decoding of claim 202, wherein said first multiplying comprises:

zeroing a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for 1 bit positions.

206 (Original). The method for fast decoding of claim 202, wherein said first multiplying comprises:

zeroing a first result, when a small number $V=\lfloor r\cdot 2^{-l}\rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shifting left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for l bit positions.

207 (Currently Amended). The method for fast decoding of claim 168, wherein said entropy decoding ~~is~~ comprises range decoding, comprising first dividing a range R with a number Total of occurrences of all symbols, to produce a prescaled range $r=\lfloor R/Total\rfloor$.

208 (Original). The method for fast decoding of claim 207, wherein said first dividing comprises shifting right said range R for $w_3=\log_2(Total)$ bit positions.

209 (Currently Amended). The method for fast decoding of claim 168, wherein said entropy decoding ~~is~~ comprises range decoding,

comprising second dividing a bottom range limit B with a prescaled range r, to produce a range correction $t=\lfloor B/r \rfloor$.

210 (Original). The method for fast decoding of claim 209, wherein said second dividing comprises:

simplified dividing said bottom range limit B with a small number $V=\lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shifting right the first result for l bit positions.

211 (Original). The method for fast decoding of claim 209, wherein said second dividing comprises:

multiplying said bottom range limit B with a first predefined number, dependent on a small number $V=\lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shifting right the first result for a sum of l and a second predefined number of bit positions, dependent on said small number V.

212 (Previously Presented). The method for fast decoding of claim 168, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients comprises reconstructing transformation coefficient *C* using a magnitude-set index *MS*, a sign *S* and a residual *R*.

213 (Original). The method for fast decoding of claim 212, wherein said entropy decoding comprises decoding said residual $R$ using inverse variable length coding (INVVLC).

214 (Previously Presented). The method for fast decoding of claim 212, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises context modeling of a transformation coefficient to be decoded, using already decoded transformation coefficients.

215 (Original). The method for fast decoding of claim 214, wherein said already decoded transformation coefficients are located north-east, north, north-west and west from the transformation coefficient to be decoded.

216 (Previously Presented). The method for fast decoding of claim 214, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises determining a mean value $\overline{MS}$ as the mean value of magnitude-set indexes $MS_i$ of said already decoded transformation coefficients.

217 (Previously Presented). The method for fast decoding of claim 216, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises limiting a maximum mean value $\overline{MS}$ by a constant $ML$ to produce a magnitude context $MC$.

218 (Original). The method for fast decoding of claim 217, wherein said entropy decoding comprises range decoding a magnitude-set index $MS$ as a current symbol $x$, using an adaptive magnitude histogram $h[MC]$.

219 (Previously Presented). The fast decoding of claim 218, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises updating of an adaptive magnitude histogram $h[MC]$ using decoded said magnitude-set index $MS$ as a current symbol $x$.

220 (Previously Presented). The method for fast decoding of claim 214, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises determining a ternary context $TC$ as the ternary code of sign values $S_i$ of said already decoded transformation coefficients.

221 (Previously Presented). The method for fast decoding of claim 220, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises translating said ternary context $TC$ into a sign context $SC$ using CTX table.

222 (Original). The method for fast decoding of claim 221, wherein said entropy decoding comprises range decoding a sign $S$ as a current symbol $x$, using an adaptive sign histogram $g[SC]$.

223 (Previously Presented). The method for fast decoding of claim 222, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises updating of an adaptive sign histogram $g[SC]$ using decoded said sign $S$ as a current symbol $x$.

224 (Previously Presented). The method for fast decoding of claim 222, wherein said estimating the probabilities of symbols within the contexts of transformation coefficients further comprises inverting less probable decoded said sign $S$ using NEG table.

225 (Currently Amended). An article of manufacture for fast encoding of input data into output compressed data, the article of manufacture comprising:

at least one non-transitory processor readable storage medium; and

instructions stored on the at least one medium;

wherein the instructions are configured to be readable from the at least one medium by at least one processor and thereby cause the at least one processor to operate so as to:

receive input data at a plurality of single-level direct subband transformers;

~~direct subband~~ transform the input data in the plurality of single-level direct subband transformers to produce transformation coefficients;

receive respective transformation coefficients directly from the plurality of single-level direct subband transformers at a plurality of processing means;

process the respective transformation coefficients received at each of the plurality of processing means ~~selected from a group consisting of: pass through transformation coefficients for lossless processing and quantize transformation coefficients for lossy processing~~ to produce processed transformation coefficients, each of the plurality of processing

means selected from a group consisting of: pass-through means for lossless processing and quantizer means for lossy processing;

estimate probabilities of symbols within contexts of transformation coefficients using the processed transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;

entropy encode the processed transformation coefficients using the probabilities of symbols within the contexts of transformation coefficients to produce encoded data; and

substantially synchronize the encoded data to produce output compressed data.

226 (Cancelled).

227 (Currently Amended). The article of manufacture for fast encoding of claim 225, further causing the at least one processor to:

~~direct subband transform input data to produce transformation coefficients;~~

~~process transformation coefficients selected from a group consisting of: pass-through transformation coefficients for lossless processing and quantize transformation coefficients for~~

~~lossy processing to produce processed transformation coefficients;~~

~~estimate probabilities of symbols within contexts of transformation coefficients using the processed transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;~~

~~entropy encode processed transformation coefficients using the probabilities of symbols within the contexts of transformation coefficients to produce encoded data;~~

substantially synchronize the encoded data to produce synchronized compressed data; and

buffer the synchronized compressed data ~~is performed~~ to produce the output compressed data.

228 (Previously Presented). The article of manufacture for fast encoding of claim 225, further causing the at least one processor to color space convert original input data to produce input data.

229 (Previously Presented). The article of manufacture for fast encoding of claim 225, wherein said direct subband transform causes the at least one processor to:

(a) direct subband transform the input data to produce transformation coefficients;

(b) direct subband transform selected transformation coefficients to produce transformed transformation coefficients; and

(c) repeat step (b) finite number of times.

230 (Original). The article of manufacture for fast encoding of claim 229, wherein said selected transformation coefficients are low-pass transformed for one-dimensional input data.

231 (Original). The article of manufacture for fast encoding of claim 229, wherein said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

232 (Previously Presented). The article of manufacture for fast encoding of claim 225, wherein said direct subband transform causes the at least one processor to:

direct filter horizontally; and

direct filter vertically.

233 (Previously Presented). The article of manufacture for fast encoding of claim 232, wherein said direct filter horizontally is different from said direct filter vertically.

234 (Previously Presented). The article of manufacture for fast encoding of claim 232, wherein at least one of said direct filter horizontally and said direct filter vertically causes the at least one processor to direct non-stationary filter.

235 (Previously Presented). The article of manufacture for fast encoding of claim 225, wherein said direct subband transform causes the at least one processor to direct filter.

236 (Previously Presented). The article of manufacture for fast encoding of claim 235, wherein said direct filter causes the at least one processor to direct non-stationary filter.

237 (Previously Presented). The article of manufacture for fast encoding of claim 236, wherein said direct non-stationary filter causes the at least one processor to perform a plurality of successive direct non-stationary cell filterings.

238 (Currently Amended). The article of manufacture for fast encoding of claim 237, wherein said direct non-stationary cell filter causes the at least one processor to:

filter using a first direct transfer function in ~~the~~ a first cycle; and

filter using a second direct transfer function in ~~the~~ a

second cycle.

239 (Original). The article of manufacture for fast encoding of claim 238, wherein said first cycle is active during horizontal filtering of each second pixel; and said second cycle is active during horizontal filtering of other pixels.

240 (Original). The article of manufacture for fast encoding of claim 238, wherein said first cycle is active during vertical filtering of each second line; and said second cycle is active during vertical filtering of other lines.

241 (Previously Presented). The article of manufacture for fast encoding of claim 238, wherein said direct non-stationary filter further causes the at least one processor to:

multiply a result of a plurality of steps that successive direct non-stationary cell filter with a first gain number to produce a first result;

multiply a result of a plurality of steps that successive direct non-stationary cell filter with a second gain number to produce a second result;

select the first result in each first cycle to produce an output sample; and

select the second result in each second cycle to produce

the output sample.

242 (Previously Presented). The article of manufacture for fast encoding of claim 238, wherein said direct non-stationary cell filter further causes the at least one processor to:

delay an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiply each even indexed delayed result with at least one first filter coefficient selected from a group of first filter coefficients to produce first results;

add together all first results to produce a third result;

add the third result with the input sample to produce a fifth result;

delay the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiply the input sample and each odd indexed delayed result with at least one second filter coefficient selected from a group of second filter coefficients to produce second results;

add together all second results to produce a fourth result;

add the fourth result with last delayed result to produce a sixth result;

output the sixth result in each first cycle; and

output the last delayed result in each second cycle.

243 (Previously Presented). The article of manufacture for fast encoding of claim 242, wherein at least one said multiply causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

244 (Previously Presented). The article of manufacture for fast encoding of claim 242, wherein said direct non-stationary filter further causes the at least one processor to:

transform the third result by first function $N_1$ to produce a seventh result;

transform the fourth result by first function $N_2$ to produce an eight result;

add the seventh result with the input sample to produce a fifth result; and

add the eight result with last delayed result to produce a sixth result.

245 (Previously Presented). The article of manufacture for fast encoding of claim 244, wherein at least one said transform causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

246 (Previously Presented). The article of manufacture for fast

encoding of claim 238, wherein said direct non-stationary cell filter further causes the at least one processor to:

delay an input sample for w samples to produce a delayed result in each first cycle;

multiply the delayed result with a second filter coefficient to produce a second result;

add the second result with the input sample to produce a fourth result;

delay the fourth result for w samples to produce the delayed result in each second cycle;

multiply the input sample with a first filter coefficient to produce a first result;

add the first result with the delayed result to produce a third result;

output the third result in each first cycle; and

output the delayed result in each second cycle.

247 (Previously Presented). The article of manufacture for fast encoding of claim 246, wherein at least one said multiply causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

248 (Previously Presented). The article of manufacture for fast encoding of claim 247, wherein said operation causes the at

least one processor to shift right data for two bit positions.

249 (Previously Presented). The article of manufacture for fast encoding of claim 247, wherein said operation causes the at least one processor to shift right data for one bit position.

250 (Previously Presented). The article of manufacture for fast encoding of claim 249, wherein said add the second result causes the at least one processor to subtract the second result from the input sample to produce a fourth result; and said add the first result causes the at least one processor to subtract the first result from the delayed result to produce a third result.

251 (Previously Presented). The article of manufacture for fast encoding of claim 238, wherein said direct non-stationary cell filter further causes the at least one processor to:

delay the input sample for w samples to produce a first delayed result in each first cycle;

delay the first delayed result for w samples to produce a second delayed result;

delay the second delayed result for w samples to produce a third delayed result;

multiply the input sample with a first filter coefficient to produce a first result;

multiply the first delayed result with a second filter coefficient to produce a second result;

multiply the second delayed result with a third filter coefficient to produce a third result;

multiply the third delayed result with a fourth filter coefficient to produce a fourth result;

add the second result with the fourth result to produce a sixth result;

add the third result with the first result to produce a fifth result;

add the sixth result with the input sample to produce an eight result;

delay the eight result for w samples to produce the first delayed result in each second cycle;

add the fifth result with the third delayed result to produce a seventh result;

output the seventh result in each first cycle; and

output the third delayed result in each second cycle.

252 (Previously Presented). The article of manufacture for fast encoding of claim 251, wherein at least one said multiply causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

253 (Previously Presented). The article of manufacture for fast encoding of claim 252, wherein said operation causes the at least one processor to shift right data for four bit positions.

254 (Previously Presented). The article of manufacture for fast encoding of claim 253, wherein:

said add the second result causes the at least one processor to subtract the second result from the fourth result to produce a sixth result; and

said add the third result causes the at least one processor to subtract the third result from the first result to produce a fifth result.

255 (Previously Presented). The article of manufacture for fast encoding of claim 225, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients causes the at least one processor to update adaptive histograms.

256 (Previously Presented). The article of manufacture for fast encoding of claim 255, wherein said updates adaptive histograms causes the at least one processor to low-pass filter probabilities selected from a group consisting of:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

257 (Previously Presented). The article of manufacture for fast encoding of claim 256, wherein said update adaptive histograms further causes the at least one processor to adapt a dominant pole during said low-pass filtering.

258 (Previously Presented). The article of manufacture for fast encoding of claim 257, wherein said adapt a dominant pole causes the at least one processor to halve value of the dominant pole in each adaptation cycle.

259 (Currently Amended). The article of manufacture for fast encoding of claim 225, wherein said entropy encode ~~is~~ comprises a range encode, causing the at least one processor to multiply a prescaled range r with a number $Q(x)$ selected from a group consisting of:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

260 (Previously Presented). The article of manufacture for fast encoding of claim 259, wherein said multiply causes the at least one processor to:

simplified multiply a small number $V=\lfloor r\cdot 2^{-l}\rfloor$ with said number $Q(x)$ to produce a first result; and

shift left the first result for l bit positions.

261 (Previously Presented). The article of manufacture for fast encoding of claim 259, wherein said multiply causes the at least one processor to shift left said number $Q(x)$ for l bit positions.

262 (Previously Presented). The article of manufacture for fast encoding of claim 259, wherein said multiply causes the at least one processor to:

zero a first result, when a small number $V=\lfloor r\cdot 2^{-l}\rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

263 (Previously Presented). The article of manufacture for fast encoding of claim 259, wherein said multiply causes the at least one processor to:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shift left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

264 (Currently Amended). The article of manufacture for fast encoding of claim 225, wherein said entropy encode ~~is~~ comprises a range encode, causing the at least one processor to:

divide a range R with a number Total of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

265 (Previously Presented). The article of manufacture for fast encoding of claim 264, wherein said divide causes the at least one processor to shift right range R for $w_3 = \log_2(Total)$ bit

positions.

266 (Previously Presented). The article of manufacture for fast encoding of claim 225, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients causes the at least one processor to:

split a transformation coefficient $C$ into a sign $S$ and a magnitude $M$;

determine a magnitude-set index $MS$ using said magnitude $M$ and a magnitude-set table; and

determine a residual $R$ using said magnitude $M$ and said magnitude-set table.

267 (Original). The article of manufacture for fast encoding of claim 266, wherein:

said magnitude-set index $MS$ is determined to be equal to a sum of a doubled position of the first nonzero bit of the highest significance and the value of the first next bit of the lower significance in a binary representation of said magnitude $M$; and

said residual $R$ is determined as the difference between said magnitude $M$ and the lower coefficient limit, equal to a value of said magnitude $M$ with all bits zeroed except the first nonzero bit of the highest significance and the first next bit

of the lower significance in a binary representation of said magnitude *M*.

268 (Previously Presented). The article of manufacture for fast encoding of claim 266, wherein said entropy encodes causes the at least one processor to encode a residual *R* using variable length coding (VLC).

269 (Previously Presented). The article of manufacture for fast encoding of claim 266, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to context model a transformation coefficient to be encoded, using already encoded transformation coefficients.

270 (Original). The article of manufacture for fast encoding of claim 269, wherein said already encoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be encoded.

271 (Previously Presented). The article of manufacture for fast encoding of claim 269, wherein said estimate the probabilities of symbols within the contexts of transformation

coefficients further causes the at least one processor to determine a mean value $\overline{MS}$ as the mean value of magnitude-set indexes $MS_i$ of said already encoded transformation coefficients.

272 (Previously Presented). The article of manufacture for fast encoding of claim 271, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to limit a maximum mean value $\overline{MS}$ by a constant $ML$ to produce a magnitude context $MC$.

273 (Previously Presented). The article of manufacture for fast encoding of claim 272, wherein said entropy encodes causes the at least one processor to range encode said magnitude-set index $MS$ as a current symbol $x$, using said adaptive magnitude histogram $h[MC]$.

274 (Previously Presented). The article of manufacture for fast encoding of claim 272, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to update an adaptive magnitude histogram $h[MC]$ using said magnitude-set index $MS$ as a current symbol $x$.

275 (Previously Presented). The article of manufacture for fast encoding of claim 269, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to determine a ternary context *TC* as the ternary code of sign values $S_i$ of said already encoded transformation coefficients.

276 (Previously Presented). The article of manufacture for fast encoding of claim 275, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to invert less probable said sign *S* using NEG table.

277 (Previously Presented). The article of manufacture for fast encoding of claim 275, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to translate said ternary context *TC* into a sign context *SC* using CTX table.

278 (Previously Presented). The article of manufacture for fast encoding of claim 277, wherein said entropy encodes causes

the at least one processor to range encode said sign $S$ as a current symbol $x$, using said adaptive sign histogram $g[SC]$.

279 (Previously Presented). The article of manufacture for fast encoding of claim 277, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to update of an adaptive sign histogram $g[SC]$ using said sign $S$ as a current symbol $x$.

280 (Currently Amended). An article of manufacture for fast decoding of input compressed data into output data, the article of manufacture comprising:

at least one <u>non-transitory</u> processor readable <u>storage</u> medium; and

instructions stored on the at least one medium;

wherein the instructions are configured to be readable from the at least one medium by at least one processor and thereby cause the at least one processor to operate so as to:

substantially synchronize input compressed data to produce synchronized compressed data;

entropy decode the synchronized compressed data using probabilities of symbols within contexts of transformation

coefficients to produce transformation coefficients;

estimate the probabilities of symbols within the contexts of transformation coefficients using the transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;

process the transformation coefficients in a plurality of processing means ~~selected from a group consisting of: pass-through transformation coefficients for lossless processing and dequantize transformation coefficients for lossy processing~~ to produce processed transformation coefficients, each of the plurality of processing means selected from a group consisting of: pass-through means for lossless processing and dequantizer means for lossy processing; ~~and~~

receive respective processed transformation coefficients directly from the plurality of processing means at a plurality of single-level inverse direct subband transformers; and

~~inverse subband~~ transform the respective processed transformation coefficients received at the plurality of single-level inverse direct subband transformers to produce output data.

281 (Cancelled).

282 (Currently Amended). The article of manufacture for fast decoding of claim 280, further causing the at least one processor to:

buffer input compressed data to produce buffered compressed data; and

substantially synchronize the buffered compressed data to produce synchronized compressed data~~;~~

~~entropy decode the synchronized compressed data using probabilities of symbols within contexts of transformation coefficients to produce transformation coefficients;~~

~~estimate the probabilities of symbols within the contexts of transformation coefficients using the transformation coefficients to produce the probabilities of symbols within the contexts of transformation coefficients;~~

~~process transformation coefficients selected from a group consisting of: pass-through transformation coefficients for lossless processing and dequantize transformation coefficients for lossy processing to produce processed transformation coefficients; and~~

~~inverse subband transform the processed transformation coefficients to produce output data~~.

283 (Previously Presented). The article of manufacture for fast decoding of claim 280, further causing the at least one

processor to color space convert the output data to produce converted output data.

284 (Previously Presented). The article of manufacture for fast decoding of claim 280, wherein said inverse subband transform causes the at least one processor to:

(a) inverse subband transform transformation coefficients to produce selected transformation coefficients;

(b) repeat step (a) finite number of times; and

(c) inverse subband transform transformation coefficients to produce the output data.

285 (Original). The article of manufacture for fast decoding of claim 284, wherein said selected transformation coefficients are low-pass transformed for one-dimensional input data.

286 (Original). The article of manufacture for fast decoding of claim 284, wherein said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

287 (Previously Presented). The article of manufacture for fast decoding of claim 280, wherein said inverse subband transform causes the at least one processor to:

inverse filter horizontally; and

inverse filter vertically.

288 (Previously Presented). The article of manufacture for fast decoding of claim 287, wherein said inverse filter horizontally is different from said inverse filter vertically.

289 (Previously Presented). The article of manufacture for fast decoding of claim 287, wherein at least one of said inverse filter horizontally and said inverse filter vertically causes the at least one processor to inverse non-stationary filter.

290 (Previously Presented). The article of manufacture for fast decoding of claim 280, wherein said inverse subband transform causes the at least one processor to inverse filter.

291 (Previously Presented). The article of manufacture for fast decoding of claim 290, wherein said inverse filter causes the at least one processor to inverse non-stationary filter.

292 (Previously Presented). The article of manufacture for fast decoding of claim 291, wherein said inverse non-stationary filter causes the at least one processor to perform a plurality of successive inverse non-stationary cell filterings.

293 (Previously Presented). The article of manufacture for fast decoding of claim 292, wherein said inverse non-stationary cell filter causes the at least one processor to:

filter using a first inverse transfer function in a first cycle; and

filter using a second inverse function in a second cycle.

294 (Previously Presented). The article of manufacture for fast decoding of claim 293, wherein said second cycle is active during horizontal filtering of each second pixel and said first cycle is active during horizontal filtering of other pixels.

295 (Previously Presented). The article of manufacture for fast decoding of claim 293, wherein said second cycle is active during vertical filtering of each second line and said first cycle is active during vertical filtering of other lines.

296 (Previously Presented). The article of manufacture for fast decoding of claim 293, wherein said inverse non-stationary filter further causes the at least one processor to:

multiply an input with a reciprocal value of a first gain number to produce a first result;

multiply an input with a reciprocal value of a second gain

number to produce a second result;

select the first result in each second cycle to produce an input sample for a plurality of steps that successive inverse non-stationary cell filter; and

select the second result in each first cycle to produce an input sample for a plurality of steps that successive inverse non-stationary cell filter.

297 (Previously Presented). The article of manufacture for fast decoding of claim 293, wherein said inverse non-stationary cell filter further causes the at least one processor to:

delay an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiply each even indexed delayed result with at least one first filter coefficient selected from a group of first filter coefficients to produce first results;

add together all first results to produce a third result;

add the third result with the input sample to produce a fifth result;

delay the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiply the input sample and each odd indexed delayed result with at least one second filter coefficient selected from a group of second filter coefficients to produce second results;

add together all second results to produce a fourth result;

add the fourth result with last delayed result to produce a sixth result;

output the sixth result in each first cycle; and

output the last delayed result in each second cycle.

298 (Previously Presented). The article of manufacture for fast decoding of claim 297, wherein at least one said multiply causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

299 (Previously Presented). The article of manufacture for fast decoding of claim 297, wherein said inverse non-stationary cell filter further causes the at least one processor to:

transform the third result by a first function $N_1$ to produce a seventh result;

transform the fourth result by a first function $N_2$ to produce an eight result;

add the seventh result with the input sample to produce a fifth result; and

add the eight result with last delayed result to produce a sixth result.

300 (Previously Presented). The article of manufacture for fast decoding of claim 299, wherein at least one said transform causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

301 (Previously Presented). The article of manufacture for fast decoding of claim 293, wherein said inverse non-stationary cell filter further causes the at least one processor to:

delay an input sample for w samples to produce a delayed result in each first cycle;

multiply the delayed result with a second filter coefficient to produce a second result;

add the second result with the input sample to produce a fourth result;

delay the fourth result for w samples to produce the delayed result in each second cycle;

multiply the input sample with a first filter coefficient to produce a first result;

add the first result with the delayed result to produce a third result;

output the third result in each first cycle; and

output the delayed result in each second cycle.

302 (Previously Presented). The article of manufacture for fast

decoding of claim 301, wherein at least one said multiply causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

303 (Previously Presented). The article of manufacture for fast decoding of claim 302, wherein said operation causes the at least one processor to shift right for one bit position.

304 (Previously Presented). The article of manufacture for fast decoding of claim 302, wherein said operation causes the at least one processor to shift right data for two bit positions.

305 (Previously Presented). The article of manufacture for fast decoding of claim 304, wherein said add the second result causes the at least one processor to subtract the second result from the input sample to produce a fourth result; and said add the first result causes the at least one processor to subtract the first result from the delayed result to produce a third result.

306 (Previously Presented). The article of manufacture for fast decoding of claim 293, wherein said inverse non-stationary cell filter further causes the at least one processor to:

delay an input sample for w samples to produce a first delayed result in each first cycle;

delay the first delayed result for w samples to produce a second delayed result;

delay the second delayed result for w samples to produce a third delayed result;

multiply the input sample with a first filter coefficient to produce a first result;

multiply the first delayed result with a second filter coefficient to produce a second result;

multiply the second delayed result with a third filter coefficient to produce a third result;

multiply the third delayed result with a fourth filter coefficient to produce a fourth result;

add the fourth result with the second result to produce a sixth result;

add the first result with the third result to produce a fifth result;

add the sixth result with the input sample to produce an eight result;

delay the eight result for w samples to produce the first delayed result in each second cycle;

add the fifth result with the third delayed result to produce a seventh result;

output the seventh result in each first cycle; and

output the third delayed result in each second cycle.

307 (Previously Presented). The article of manufacture for fast decoding of claim 306, wherein at least one said multiply causes the at least one processor to perform an operation selected from a group consisting of: shift and bit remap.

308 (Previously Presented). The article of manufacture for fast decoding of claim 307, wherein said operation causes the at least one processor to shift right data for four bit positions.

309 (Previously Presented). The article of manufacture for fast decoding of claim 308, wherein said add the fourth result causes the at least one processor to subtract the fourth result from the second result to produce a sixth result; and said add the first result causes the at least one processor to subtract the first result from the third result to produce a fifth result.

310 (Previously Presented). The article of manufacture for fast decoding of claim 280, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients causes the at least one processor to update adaptive histograms.

311 (Previously Presented). The article of manufacture for fast

decoding of claim 310, wherein said update adaptive histograms causes the at least one processor to low-pass filter probabilities selected from a group consisting of:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding said current symbol x.

312 (Previously Presented). The article of manufacture for fast decoding of claim 311, wherein said update adaptive histograms further causes the at least one processor to adapt a dominant pole during said low-pass filtering.

313 (Previously Presented). The article of manufacture for fast decoding of claim 312, wherein said adapt a dominant pole causes the at least one processor to halve value of the dominant pole in each adaptation cycle.

314 (Currently Amended). The article of manufacture for fast decoding of claim 280, wherein said entropy decode ~~is~~ comprises a range decode, causing the at least one processor to multiply a prescaled range r with a number $Q(x)$ selected from a group consisting of:

a number $U(x)$ of occurrences of all symbols preceding a

current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

315 (Previously Presented). The article of manufacture for fast decoding of claim 314, wherein said multiply causes the at least one processor to:

simplified multiply a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shift left the first result for l bit positions.

316 (Previously Presented). The article of manufacture for fast decoding of claim 314, wherein said multiply causes the at least one processor to shift left said number $Q(x)$ for l bit positions.

317 (Previously Presented). The article of manufacture for fast decoding of claim 314, wherein said multiply causes the at least one processor to:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number

higher or equal 3;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for 1 bit positions.

318 (Previously Presented). The article of manufacture for fast decoding of claim 314, wherein said multiply causes the at least one processor to:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shift left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for 1 bit positions.

319 (Currently Amended). The article of manufacture for fast decoding of claim 280, wherein said entropy decode ~~is~~ comprises a range decode, causing the at least one processor to divide a range R with a number Total of occurrences of all symbols, to

produce a prescaled range $r = \lfloor R/Total \rfloor$.

320 (Previously Presented). The article of manufacture for fast decoding of claim 319, wherein said divide causes the at least one processor to shift right said range R for $w_3 = \log_2(Total)$ bit positions.

321 (Currently Amended). The article of manufacture for fast decoding of claim 280, wherein said entropy decode ~~is~~ comprises a range decode, causing the at least one processor to divide a bottom range limit B with a prescaled range r, to produce a range correction $t = \lfloor B/r \rfloor$.

322 (Previously Presented). The article of manufacture for fast decoding of claim 321, wherein said divide causes the at least one processor to:

simplified divide said bottom range limit B with a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shift right the first result for l bit positions.

323 (Previously Presented). The article of manufacture for fast decoding of claim 321, wherein said divide causes the at least one processor to:

multiply said bottom range limit B with a first predefined number, dependent on a small number $V=\lfloor r\cdot 2^{-l}\rfloor$ to produce a first result; and

shift right the first result for a sum of 1 and a second predefined number of bit positions, dependent on said small number V.

324 (Previously Presented). The article of manufacture for fast decoding of claim 280, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients causes the at least one processor to reconstruct transformation coefficient *C*, using a magnitude-set index *MS*, a sign *S* and a residual *R*.

325 (Previously Presented). The article of manufacture for fast decoding of claim 324, wherein said entropy decode causes the at least one processor to decode said residual *R* using inverse variable length coding (INVVLC).

326 (Previously Presented). The article of manufacture for fast decoding of claim 324, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to

context model a transformation coefficient to be decoded, using already decoded transformation coefficients.

327 (Original). The article of manufacture for fast decoding of claim 326, wherein said already decoded transformation coefficients are located north-east, north, north-west and west from the transformation coefficient to be decoded.

328 (Previously Presented). The article of manufacture for fast decoding of claim 326, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to determine a mean value $\overline{MS}$ as the mean value of magnitude-set indexes $MS_i$ of said already decoded transformation coefficients.

329 (Previously Presented). The article of manufacture for fast decoding of claim 328, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to limit a maximum mean value $\overline{MS}$ by a constant $ML$ to produce a magnitude context $MC$.

330 (Previously Presented). The article of manufacture for

fast decoding of claim 329, wherein said entropy decode causes the at least one processor to range decode said magnitude-set index *MS* as a current symbol *x*, using said adaptive magnitude histogram $h[MC]$.

331 (Previously Presented). The article of manufacture for fast decoding of claim 330, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to update an adaptive magnitude histogram $h[MC]$ using said magnitude-set index *MS* as a current symbol *x*.

332 (Previously Presented). The article of manufacture for fast decoding of claim 326, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to determine a ternary context *TC* as the ternary code of sign values $S_i$ of said already decoded transformation coefficients.

333 (Previously Presented). The article of manufacture for fast decoding of claim 332, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to

translate said ternary context $TC$ into a sign context $SC$ using CTX table.

334 (Previously Presented). The article of manufacture for fast decoding of claim 333, wherein said entropy decode causes the at least one processor to range decode sign $S$ as a current symbol $x$, using an adaptive sign histogram $g[SC]$.

335 (Previously Presented). The article of manufacture for fast decoding of claim 334, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to update an adaptive sign histogram $g[SC]$ using decoded sign $S$ as a current symbol $x$.

336 (Previously Presented). The article of manufacture for fast decoding of claim 334, wherein said estimate the probabilities of symbols within the contexts of transformation coefficients further causes the at least one processor to invert less probable decoded sign $S$ using NEG table.

337-448 (Cancelled).

449 (Currently Amended). An article of manufacture for fast encoding of input data into output compressed data comprising a at least one non-transitory processor readable storage medium for storing a computer program of instructions configured to be readable by at least one processor for instructing the at least one processor to execute a computer process for performing the method of claim 113.

450 (Currently Amended). An article of manufacture for fast decoding of input compressed data into output data comprising a at least one non-transitory processor readable storage medium for storing a computer program of instructions configured to be readable by at least one processor for instructing the at least one processor to execute a computer process for performing the method of claim 168.

451-452 (Cancelled).